Multimedia Tools and Applications (2020) 79:26345–26368
https://doi.org/10.1007/s11042-020-09256-z

Fractal image compression with adaptive quadtree partitioning and non-linear affine map

Utpal Nandi[1]

Received: 28 June 2019 / Revised: 4 June 2020 / Accepted: 24 June 2020 /
Published online: 14 July 2020

Abstract
Fractal image compression techniques are now very popular for its high compression rates and resolution independence property. However, the qualities of decoded images of the existing techniques are not satisfactory. An adaptive partitioning scheme can improve the image quality significantly. These existing adaptive techniques use linear affine maps during encoding that have limited pixel intensity approximation ability. In order to increase the image quality further, non-linear affine maps can be used that generalizes the pixel intensity approximation and generates much better approximation. Here, a fractal based technique for image compression using non-linear contractive affine maps has been proposed that applies adaptive quadtree partitioning to partition image in a context dependent way to enhance decoded image quality. The technique partitions twice an image to be compressed to obtain collection of ranges and domains and finds the highest matching non-linear affine transformed domain of each range. The corresponding affine parameters are kept in the compressed file. However, a range may be broken into sub-ranges using adaptive quadtree partitioning for unavailability of enough matching domains and repeat the same on those. The comparative results show that the proposed technique greatly improves the decoded image quality than existing techniques and also maintains the high compression ratios. Two variants have also been proposed that improve compression ratio of the proposed technique without any degradation of image quality using loss-less coding.

Keywords Fractal compression · Adaptive quadtree partition · Affine map · Non-linear affine map · Contractive transform · Compression ratio · Image quality

1 Introduction

Multimedia uses a combination of image, audio, text, video and animation and its security [19, 20, 49] becomes one of the interesting areas of research today. Again, the sizes of multimedia data are much larger than classical media tools. As a result, it takes huge

✉ Utpal Nandi
nandi.3utpal@gmail.com

[1] Department of Computer Science, Vidyasagar University, Midnapore, West Bengal, Pin-721102, India

storage space and high bandwidth to transfer over the network. Therefore, size reduction of multimedia data is also very important. Moreover, image size deduction that is the basis of multimedia data also depicts its significance. The compression techniques [32] provide a good solution to these problems. Several techniques and standards already developed to reduce the image size of multimedia data. The compression techniques are lossless or lossy [32]. Mainly the lossy techniques are applied for image coding like DCT based JPEG [39] and DWT based wavelet [2, 7, 8, 35, 36] image coding. The DCT based JPEG coding is resolution dependent and is not efficient to produce good quality highly compressed image. Fractal image compression (FIC) [9] plays here a significant role since fractal compressed image does not depend on resolution and enlarged portions of image are free of blocky areas. The decoding time of the technique is very fast. The fractal coding represents image as a set of affine maps. However, the qualities of decoded images of the existing fractal compression techniques are not satisfactory. The most of the existing techniques use linear affine maps [9, 32] during encoding that have limited pixel intensity approximation ability. In order to increase the image quality further, non-linear affine maps (second order) [16] can be used that generalizes the pixel intensity approximation and generates much better approximation. However, the transformation should be contractive to use it in fractal compression. Zhao et al. [50] applied non-linear affine maps on fractal compression technique with quadtree partitioning (QP) scheme [9, 32]. The partitioning scheme of fractal compression is an important factor to enhance reconstructed image quality. However, the QP does not depend on the pixels of the image block so that partitioned sub-blocks have matching portions. An adaptive quadtree partitioning (AQP) scheme [29] can be applied to solve this problem that splits image block in such a way that it enhance the possibility of finding self-matching portions. Therefore, a FIC technique has been proposed that applies non-linear affine maps and AQP scheme to enhance the quality of reconstructed image. The technique partitions an image to be compressed into a number of non-overlapping ranges and its double size overlapping domains and finds the highest matching non-linear affine transformed domain of each range. The corresponding affine parameters are kept in the compressed file. However, a range may be broken into sub-ranges using AQP for unavailability of enough matching domains and repeat the same on those. The decompression process takes the compressed file and a gray scale black image. It iterates affine transforms kept in the compressed file on the black image to reconstruct the decoded image. The related works are discussed in Section 2. The detail of non-linear affine map and it's contractive transformation property are well discussion in Section 3. The applied partitioning strategy is explained in Section 4. After that, a FIC using non-linear second order contractive affine maps and AQP scheme is proposed and discussed in detail in Section 5. Two variants of the same have also been proposed in section 6 that apply loss-less coding [27, 28] to enhance the compression ratio. Then, the experimental results are shown and analyzed in Section 7. Conclusion is made in Section 8. Finally, the references are given.

2 Related work

Barnsely and Hurd [3] first introduced the usage of IFS in image coding. The IFS based FIC technique was not applicable on natural images. After that Jacquin [14] presented partitioned IFS for FIC that worked on natural images also. However, the technique suffered from huge encoding time. Several research works were continuing to enhance the performance of the technique. The image block classification strategy of fractal coding is one of

the main aspect of research to enhance performance. Fu and Zhu [10] proposed DCT based classification of image block to reduce the number of range-domain pair searching where Wang and Zhang [43] applied DWT for the same in fractal compression. Wang and Zheng [44] presented a FIC that classifies and sorts image blocks using correlation coefficients. Xing et al.[47] and Battacharayaet et al. [4] presented two different classification schemes for FIC that grouped domains in two level hierarchies to speed up the domain searching procedures. A different classification strategy for AQP based technique proposed by Nandi and Mandal [31] that also maintained two level hierarchies and then, Nandi [38] suggested an improved scheme for the same. Another aspect of research to enhance performance of fractal coding is partitioning scheme. Nandi and Mandal [29] suggested adaptive quadtree partitioning scheme where partitioning points were determined in such a way that increased the possibility of finding matching portions of images. Two HV partitioning based FIC [26] were also proposed by them to speed up the HV partitioning scheme. Nandi and Mandal [30] also applied adaptive quadtree partitioning scheme with archetype classification and performance was farther improved by using loss-less encoding. No-search fractal techniques are also suggested by Wang et al. [42] using fitting plane and Gupta et al. [11] in DCT domain. The Huber loss function, genetic algorithm, neural network and fuzzy logic were also applied to fractal coding. Jeng et al. [15, 24] introduced fractal coding using Huber loss function. Bobde et al. [6] used a variable genetype representation of IFS to improve fractal coding. Ai-Jawfi et al. [1] applied neural network methods on fractal encoding and Lakshmi [18] implemented a fractal image coding technique using neural network for MRI image also. Nodehi et al. [33] improved fractal method by using intelligent fuzzy approach that performed better than genetic algorithm based methods. Gupta et al. [12] made a comparative analysis of edge based fractal coding for various frequency domains using nearest neighbor technique. The technique exercised FFT, DCT and real DCT on edge based fractal coding. Liu et al. [23] suggested a FIC for agricultural images and they [21] also presented a fast FIC using collected fundamental error. Liu et al. [22] applied distance grouping to make encoding more faster of FIC. Roy et al. [34] presented a fractal compression technique that calculated scaling parameters of affine maps with upper bound to reduce its complexity during encoding. All the techniques used linear affine maps at the time of encoding with limited pixel intensity approximation ability. To improve the image quality further, non-linear affine map has been used in proposed technique to generalize better the pixel intensity approximation.

3 Non-linear affine map and its contractive transformation property

The general form of affine map used in FIC techniques is in the form of (1) where (x, y) denotes the co-ordinate of a pixel to be mapped and z is the gray value. The terms a_1, a_2, a_3, a_4, g_1 and g_2 are geometric parts. The function f(z) represents contrast factor. The affine map applied in most of the fractal compression techniques is only a special case of (1) where f(z) is obtained by using (2) and c and o represent scaling factor and brightness offset respectively.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & 0 \\ 0 & 0 & f(z) \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ 0 \end{bmatrix} \tag{1}$$

$$f(z) = cz + o \tag{2}$$

Here, f(z) is linear and contractive mapping condition does not depend on the grey value of the domain pixels. Therefore, any arbitrary image can be used as the initial image during decoding process. But, its pixel intensity approximation is very limited and decoded image quality is poor. To reduce this problem, a non-linear affine map is used in the proposed technique that is in the form of (3). It generalizes the pixel intensity approximation to provide a better approximation where f(z) is calculated by (4).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & 0 \\ 0 & 0 & f(z) \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ 0 \end{bmatrix} \tag{3}$$

$$f(z) = c_1 z^2 + c_2 z + o \tag{4}$$

This affine map can be applied in FIC if it is contractive. Any affine map in the form of (3) is contractive if $\left|f'(z)\right| \leq \text{h}, 0 \leq h < 1$ is satisfied and $\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ is contractive where $f'(z)$ is the first order derivative and $\left|f'(z)\right|$ is the absolute value of $f'(z)$. In compression process, $\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ scales down the size of domain block into as small size block as range and therefore, it is contractive most of the time. Now, if $\left|f'(z)\right| \leq \text{h}, 0 \leq h < 1$ is satisfied, the affine map of (3) can be used in compression. Let us consider any two vectors $\vec{p_1} = (x_1, x_2, x_3)$, $\vec{p_2} = (y_1, y_2, y_3)$ and its corresponding affine transforms $\vec{q_1} = (X_1, X_2, X_3)$, $\vec{q_2} = (Y_1, Y_2, Y_3)$ respectively. Then, we have

$$\left(\vec{q_1} - \vec{q_2}\right) = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & 0 \\ 0 & 0 & f(x_3) - f(y_3) \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \tag{5}$$

where $D_1 = X_1 - Y_1$, $D_2 = X_2 - Y_2$, $D_3 = X_3 - Y_3$ and $d_1 = x_1 - y_1$, $d_2 = x_2 - y_2$, $d_3 = x_3 - y_3$ (say). As $\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ is set contractive, there exist a factor (say h_1) where $0 \leq h_1 < 1$ that satisfy

$$\sqrt{{D_1}^2 + {D_2}^2} \leq h_1 \sqrt{{d_1}^2 + {d_2}^2} \tag{6}$$

From Lagrange mean-value theorem, $D_3 = X_3 - Y_3 = f'(x_3) - -f'(y_3) = f'(\xi)(x_3 - y_3) = f'(\xi)$
$(d_3), \xi \epsilon (x_3, y_3)$.

$$Therefore, \quad \sqrt{{D_3}^2} = \left|f'(\xi)\right| \sqrt{{d_3}^2} \leq h_2 \sqrt{{d_3}^2}, \quad 0 \leq h_2 < 1 \tag{7}$$

Let us consider the maximum of factor h_1 and h_2 is h_{max}. Then,

$$\sqrt{{D_1}^2 + {D_2}^2 + {D_3}^2} \leq h_{max} \sqrt{{d_1}^2 + {d_2}^2 + {d_3}^2} \tag{8}$$

is satisfied and the affine map in the form of (3) is contractive. Now, it can be applied in FIC techniques and decoding process converges into a fixed image with in a finite number of iterations.

4 The adaptive quadtree partitioning (AQP) scheme

In QP scheme [9, 32], partitioning of image block is context independent and does not share self-similar structures. Therefore, the quality of decoded image is not so good. This limitation is eliminated in AQP scheme [29] where splitting of the image is context dependent to share some self-similar structure. Let, a $M \times N$ range block holds the pixels $P_{k,l}$ for $0 \leq k < M$ and $0 \leq l < N$, with left-top point (a, b) as depicted in Fig. 1a.

In the first step of AQP, the range block is divided in horizontal direction. To do so, raw-wise pixel sum (S_k) is calculated for each pixel raw k, $a \leq k \leq a + M - 1$ using (9). Then, the absolute differences between k^{th} and $(k+1)^{th}$ row sum is computed for each pixel raw k, $a \leq k \leq a + M - 1$ as $|S_k - S_{k+1}| = \left| \left(\sum_{l=b}^{b+N-1} P_{k,l} - \sum_{l=b}^{b+N-1} P_{k+1,l} \right) \right|$. Then, a linear biasing function MIN (k, M-k-1) is to compute the biased horizontal differences (HD_k) between k^{th} and $(k+1)^{th}$ row as given in (10). Next, a horizontal line k (say M_p) is selected that maximize HD_k for $a \leq k \leq a + M - 1$. The horizontal line M_p splits range block into R_1 and R_2 as in Fig. 1b.

$$S_k = \sum_{l=b}^{b+N-1} P_{k,l} \tag{9}$$

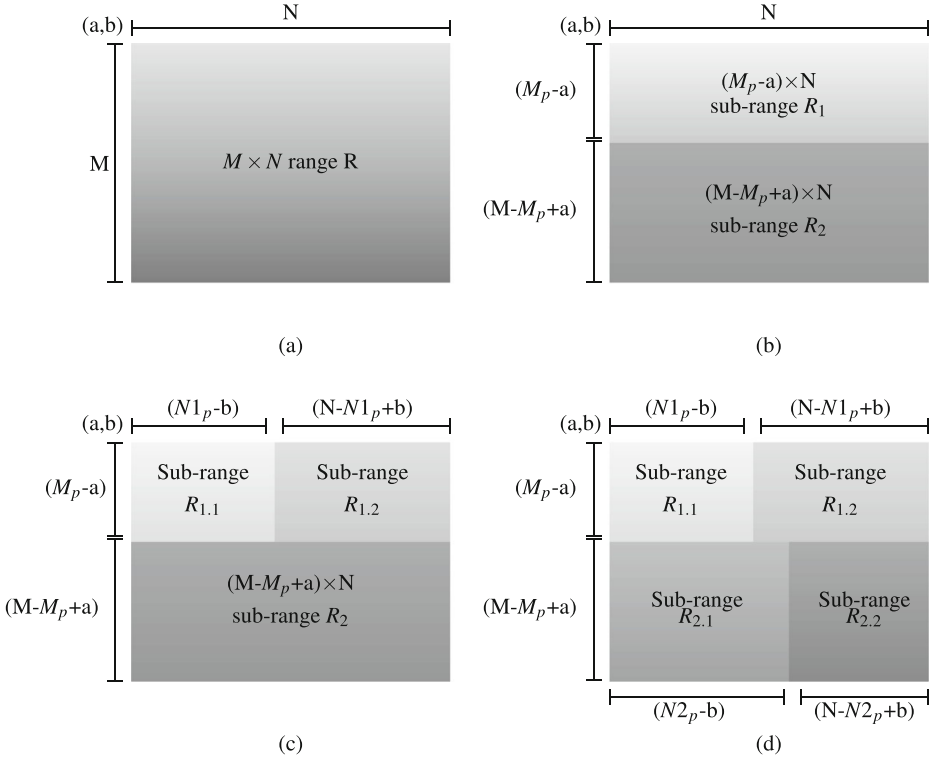

Fig. 1 The Adaptive Quadtree Partitioning (AQP) scheme

$$HD_k = MIN(k, M-k-1)\left|\left(\sum_{l=b}^{b+N-1} P_{k,l} - \sum_{l=b}^{b+N-1} P_{k+1,l}\right)\right|, \tag{10}$$

for each pixel row k, $a \le k \le a+M-1$.

In second step, R_1 is divided in vertical direction. The vertical pixel sum (S_l) is calculated for each column l, $b \le l \le b+N-1$ of R_1 using (11). Then, the absolute differences between l^{th} and $(l+1)^{th}$ column sum i.e. $|S'_l - S'_{l+1}| = \left|\left(\sum_{k=a}^{M_p-1} P_{k,l} - \sum_{k=a}^{M_p-1} P_{k,l+1}\right)\right|$ is computed for each pixel column l, $b \le l \le b+N-1$ and the liner biasing function MIN(l-b, N-l+b-1) is used to compute the biased differences as given in (12). After that, a vertical line l (say $N1_p$) is determined that maximize $VD1_l$ for $b \le l \le b+N-1$. The vertical line $N1_p$ splits R_1 into $R_{1.1}$ and $R_{1.2}$ as depicted in Fig. 1c.

$$S'_l = \sum_{k=a}^{M_p-1} P_{k,l} \tag{11}$$

$$VD1_l = MIN(l-b, N-l+b-1)\left|\left(\sum_{k=a}^{M_p-1} P_{k,l} - \sum_{k=a}^{M_p-1} P_{k,l+1}\right)\right|, \tag{12}$$

for each pixel row l, $b \le l \le b+N-1$.

In the last step, R_2 is divided in vertical direction by similar way. The calculated absolute value of successive differences is $\left|(\sum_{l=M_p}^{a+M-1} P_{k,l} - \sum_{l=M_p}^{a+M-1} P_{k,l+1}\right|$ for each pixel column l, $b \le l \le b+N-1$. and the biased difference is given in (13). After that, a vertical line l (say $N2_p$) is found that maximize $VD2_l$ for $b \le l \le b+N-1$. The vertical line $N2_p$ divides R_2 into $R_{2.1}$ and $R_{2.2}$ as illustrated in Fig. 1d.

$$VD2_l = MIN(l-b, N-l+b-1)\left|(\sum_{l=M_p}^{a+M-1} P_{k,l} - \sum_{l=M_p}^{a+M-1} P_{k,l+1}\right|, \tag{13}$$

for each pixel row l, $b \le l \le b+N-1$.

Finally, $M_p \times N1_p$ $R_{1.1}$, $M_p \times (N-N1_p)$ $R_{1.2}$, $(M-M_p) \times N2_p$ $R_{2.1}$ and $(M-M_p) \times (N-N2_p)$ $R_{2.2}$ sub-ranges are created with left-top points (a,b), (a,$N1_p$), (M_p,b) and (M_p, $N2_p$) respectively. An example of the sheme is given in Section 4.1.

4.1 Example of AQP scheme

Consider an 8 × 8 image block (Fig. 2a). First, the pixel sum of eight rows are calculated as S_1=1580, S_2=1580, S_3=1700, S_4=1660, S_5=1610, S_6=770, S_7=805 and S_8= 810 respectively. The biased horizontal differences between successive rows 1&2, 2&3, 3&4, 4&5, 5&6, 6&7 and 7&8 are calculated as min(0,7)|1580−1580|=0, min(1,6)|1580−1700|=120, min(2,5)|1700−1660|=80, min(3,4)|1660−1610|=150, min(4,3)|1610−770|=2520, min(5, 2) |770 − 805|=70 and min(6,1)|805 − 810|=5 respectively. The maximum difference is found between 5^{th} and 6^{th} row. Therefore, the block is partitioned horizontally between 5^{th} and 6^{th} row to form two sub-blocks R_1 and R_2 (Fig. 2b).

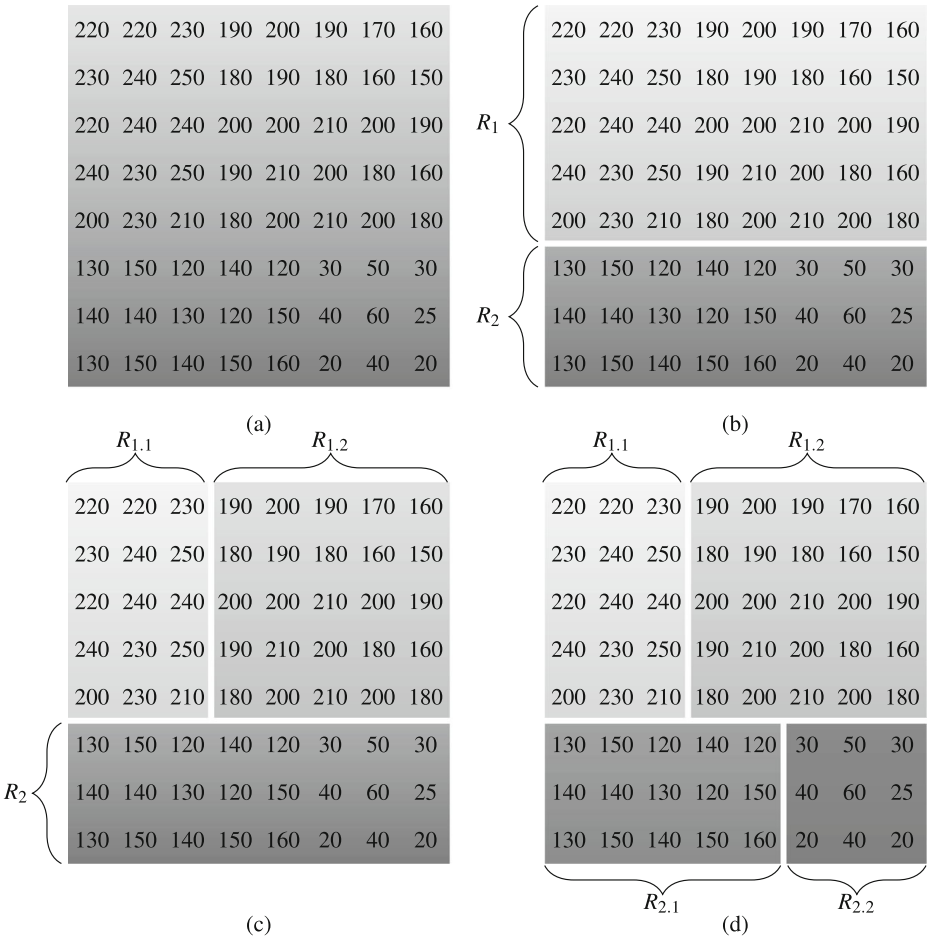

Fig. 2 Example of AQP scheme **a** An image block of size 8 × 8 **b** After horizontal partitioning of the block into sub-blocks R_1 and R_2 **c** After vertical partitioning of R_1 into $R_{1.1}$ and $R_{1.2}$ **d** After vertical partitioning of R_2 into $R_{2.1}$ and $R_{2.2}$

After that, the calculation of vertical pixel sum of eight columns of sub-block R_1 are done as S'_1=1110, S'_2=1160, S'_3=1180, S'_4=940, S'_5=1000, S'_6=990, S'_7=910 and S'_8=840 respectively. The biased vertical differences between 1&2, 2&3, 3&4, 4&5, 5&6, 6&7 and 7&8 columns are calculated as min(0,7)|1110 − 1160|=0, min(1,6)|1160 − 1180|=20, min(2,5)|1180 − 940|=280, min(3,4)|940 − 1000|=180, min(4,3)|1000 − 990|=30, min(5,2)|990 − 910|=160 and min(6,1) |910 − 840|=70 respectively. As a result, the sub-block R_1 is partitioned vertically between 3^{rd} and 4^{th} column to form two sub-blocks $R_{1.1}$ and $R_{1.2}$ as shown in Fig. 2c since the maximum difference is found between 3^{rd} and 4^{th} column.

Finally, the calculation of vertical pixel sum of of eight columns of sub-block R_2 are done as S''_1=400, S''_2=440, S''_3=390, S''_4=410, S''_5=430, S''_6=90, S''_7=150 and S''_8=75 respectively. The biased vertical differences between successive columns are calculated as min(0,7)|400 − 440|=0, min(1,6)|440−390|=50, min(2,5)|390−410|=40, min(3,4)|410−430|=60, min(4,3) |430 − 90|=990, min(5,2)|90 − 150|=120 and min(6,1)|150 − 75|=75 respectively. The

maximum difference is found between 5^{th} and 6^{th} column and the sub-block R_2 is divided vertically into two sub-blocks $R_{2.1}$ and $R_{2.2}$ between same as illustrated in Fig. 2d.

5 The proposed technique

The main features of the proposed technique are the use of non-linear affine maps instead of linear affine maps to produce better pixel intensity approximation and AQP instead of QP scheme to partition image in a context dependent way to enhance decoded image quality. The proposed technique is termed as FIC with AQP and non-linear affine maps (FIC-AQP-NAM). In case of linear affine map, finding optimum affine map requires minimization of the sum of squared error between range and mapped domain. The error value is calculated by using (14). The optimal values of the parameters c and o are obtained when the partial derivatives of E_1 w. r. t c and o are all zero as given in the following (15) and (16) respectively.

$$E_1 = \sum_{i=1}^{n} \{Z_i - f(z_i)\}^2 = \sum_{i=1}^{n} (Z_i - cz_i - o)^2 \tag{14}$$

$$\frac{\partial E_1}{\partial c} = -2 \sum_{i=1}^{n} (Z_i - cz_i - o) z_i = 0 \tag{15}$$

$$\frac{\partial E_1}{\partial o} = -2 \sum_{i=1}^{n} (Z_i - cz_i - o) = 0 \tag{16}$$

These equations are easily solved and the optimal parameters are obtained as given below:

$$c = \frac{\{nS(Zz) - S(Z)S(z)\}}{\{nS(z^2) - S(z)^2\}}$$
$$o = \{S(Z) - cS(z)\}$$

where $S(M) = \sum_{i=1}^{n} m_i$, $S(MN) = \sum_{i=1}^{n} m_i n_i$ and $S(M^p N^q) = \sum_{i=1}^{n} {m_i}^p {n_i}^q$. However, the proposed technique applies a non-linear affine map instead of linear affine map to produce better pixel intensity approximation. To find optimum affine map of the form of (3) from a range to domain, the sum of squared error between range and mapped domain is minimized here also. The error value is calculated by using (17). The optimal values of the parameters c_1, c_2, o are obtained when the partial derivatives of E_2 w.r.t c_1, c_2, o are all zero as given in the following (18), (19) and (20) respectively.

$$E_2 = \sum_{i=1}^{n} \{Z_i - f(z_i)\}^2 = \sum_{i=1}^{n} (Z_i - c_1 {z_i}^2 - c_2 z_i - o)^2 \tag{17}$$

$$\frac{\partial E_2}{\partial c_1} = -2 \sum_{i=1}^{n} (Z_i - c_1 {z_i}^2 - c_2 z_i - o) {z_i}^2 = 0 \tag{18}$$

$$\frac{\partial E_2}{\partial c_2} = -2 \sum_{i=1}^{n} (Z_i - c_1 {z_i}^2 - c_2 z_i - o) z_i = 0 \tag{19}$$

$$\frac{\partial E_2}{\partial o} = -2\sum_{i=1}^{n}(Z_i - c_1 {z_i}^2 - c_2 z_i - o) = 0 \tag{20}$$

These equations are easily solved and the optimal parameters are obtained as

$$c_1 = \frac{\{S(Zz) - S(Z)S(z)\} + c_2\{S(z)^2 - S(z^2)\}}{\{S(z^2)S(z) - S(z)^3\}}$$

$$c_2 = \frac{\{S(Zz) - S(Z)S(z)\}\{S(z^2)^2 - S(z^4)\} - \{S(Zz^2)^2 - S(z^2)S(Z)\}\{S(z^2)S(z) - S(z^3)\}}{\{S(z^2)S(z) - S(z^3)\}^2 - \{S(z)^2 - S(z^2)\}\{S(z^2)^2 - S(z^4)\}}$$

$$o = S(Z) - c_2 S(z) - c_1 S(z^2)$$

where $S(M) = \sum_{i=1}^{n} m_i$, $S(MN) = \sum_{i=1}^{n} m_i n_i$ and $S(M^p N^q) = \sum_{i=1}^{n} m_i{}^p n_i{}^q$. These calculated parameters must satisfy the contractivity condition $\left|f'(z_i)\right| = |c_1 + 2c_2 z_i| < 1$ i.e. $-1 < c_1 + 2c_2 z_i < 1$ where z_i represents pixel value of affine transformed domain block $AM_k(DMN_j)$. It is very time consuming task to check every pixel of the block and therefore, its equivalent two (21) and (22) are used for the same as follows:

$$-1 < c_1 + 2c_2 z_{max} < 1 \tag{21}$$

$$-1 < c_1 + 2c_2 z_{min} < 1 \tag{22}$$

where z_{max} and z_{min} represent maximum and minimum pixel values of the block $AM_k(DMN_j)$ respectively. The next optimal parameters are computed sometime if optimal parameters do not satisfy the contractive condition. The optimal values of c_1, c_2 and o are floating point numbers that are quantized to ensure a compact encoding of the affine map. Instead of these three parameters, the position of domain (g_1, g_2) and information about rotation with flip operation types of the affine map AM_k are quantized and stored in the compressed file. The affine map with quantized parameters is not the optimal map, but a better one. The details of compression and decompression algorithm of FIC-AQP-NAM technique are given in Sections 5.1 and 5.2 respectively.

5.1 Compression algorithm

The compression steps are stated in Algorithm 1. First, an image file (say IMG) and a tolerance level QLTY (Image quality) is taken where lower QLTY means higher image quality. The minimum size (say MIN) of range block is assigned as 2×2. Next, the consecutive domain latency known as domain density (DD) is also assigned. Initially, the algorithm subtracts 128 from all pixels of the image IMG. Then, the whole image is partitioned into 8×8 non-overlapping ranges (RNG_i), $1 \leq i \leq NR$ (Number of ranges) that are marked as uncovered and a domain pool is created from input image by dividing input image into domain blocks where size of domains are at least double than range blocks to satisfy contractive transform property. These domains may overlap with each others. The domain pool is classified into 24 classes (CLS_i) by using Fisher's classification [9, 32]. Now, for each uncovered range RNG_i, its corresponding class CLK_k is found by the same way and a domain DMN_k is selected from the class CLK_k only such that AM_k covers best RNG_k

than others i.e. that minimize RMS error $DIFF_{RMS}(RNG_k, AM_k\ (DMN_k))$. If the parameters c_1, c_2 and o of AM_k do not satisfy the contractivity condition, the next optimal parameters are computed. Again, if the RMS error is is below QLTY or it is MIN range block, then the parameters c_1, c_2 and o are quantized with other two parameters i.e. position of domain (g_1, g_2) and information of rotation with flip operation type of the affine map AM_k. In this way, if a domain is found for range R_k that is similar enough, the parameters of the AM_k is kept to the compressed file and this range is marked as covered. Otherwise, the R_k is broken into four range blocks RNG_{k1}, RNG_{k2}, RNG_{k3} and RNG_{k4} using AQP scheme and these belongs to uncovered range blocks. This process terminates when all range blocks are covered and the algorithm finally produces the compressed image IMG_c.

Algorithm 1 The FIC-AQP-NAM compression algorithm.

Require: A gray scale rectangular image (F)

1: **Input an image IMG. Set the tolerance level of quality and minimum range size as QLTY and MIN respectively. Set also domain density as DD.**
2: **Subtract 128 from all pixels of the image IMG.**
3: **Partition IMG into** 8×8 **ranges** (RNG_i)**,** $1 \leq i \leq NR$ **(Number of ranges) and these are labeled as uncovered.**
4: **Create a domain pool maintaining domain density DD that may overlap each other.**
5: **Classify all domains into 24 classes** $(CLS_i, 1 \leq i \leq 24)$ **by using Fisher's classification.**
6: **While (Range blocks** RNG_k **exist that do not covered yet), do**
    6.1: Find the class (CLS_k) of range (RNG_k) by using Fisher's classification and set CLS_j Null.
    6.2:Search the domain DMN_k from the class CLS_k that has same class as RNG_k and its corresponding non- linear affine map AM_k that cover range RNG_k well than others i.e. that minimize RMS error $DIFF_{RMS}(RNG_k, AM_k\ (DMN_k))$.
    6.3:If (NOT $(-1 < c_1 + 2c_2z_{max} < 1$ AND $-1 < c_1 + 2c_2z_{min} < 1))$, then
    Compute next optimal parameters c_1, c_2 and o.
    GOTO Step 5.3.
    End If;
    6.4: If ($(DIFF_{RMS}(RNG_k, AM_k\ (DMN_k)) <$ QLTY) OR (Size of $RNG_k) \leq$ MIN),then
        6.4.1: Quantize the parameters c_1, c_2, o,position of domain (g_1, g_2) and information of rotation with flip operation type of the affine map AM_k.
        6.4.2: The parameters of the affine map AM_k is written to the compressed file and RNG_k is labeled as covered.
        6.4.3: Else, partition RNG_k into four sub- blocks RNG_{k1}, RNG_{k2}, RNG_{k3}, RNG_{k4} using AQP scheme and label these as uncovered.
    End if;
    End while;
7: **The fractal compressed image file is produced containing a list of affine maps. (say** IMG_C**).**
8: **Stop.**

5.2 Decompression algorithm

The steps of FIC-AQP-NAM decompression algorithm are given in Algorithm 2. Initially, take an FIC-AQP-NAM compressed file IMG_C. The number of iteration (i.e. number of time affine maps will be applied) is set as I_n and scale factor (say S_f) scales the image by a factor of S_f. After that, the compressed image size ($size_x \times size_y$) is read and domain density DD is also taken from the input file. Next, the desired reconstructed image size is calculated by (23) and (24).

$$dsize_x = S \times size_x \tag{23}$$

$$dsize_y = S \times size_y \tag{24}$$

In the next step, the FIC-AQP-NAM decompression algorithm initializes the domain information similar with compression process and takes a $dsize_x \times dsize_y$ black image (IMG_R). Here, any arbitrary image cannot be taken as the initial image during decoding process. Because, the non-linear affine map has second order term and the contractive condition of each transform depends on the gray values of domains. This limitation can be overcome by adjusting gray values of image after each iteration such that gray values remain within -128 and 127 that is for each pixel value z_i of the image IMG_R, if $z_i < -128$, set z_i = -128 and if $z_i > 127$, set z_i = 127. Then, the decoder reads all affine maps of IMG_C. Now, all these affine maps of IMG_C are applied over IMG_R iteratively and a completely new image is produced. This produced image now acts as IMG_R for the next iteration. This is repeated I_n times and produces a $dsize_x \times dsize_y$ IMG_R as decompressed image. Finally, the algorithm smoothes the transition between adjacent boundaries of ranges of IMG_R by pixel averaging.

Algorithm 2 TheFIC-AQP-NAM decompression algorithm.

Require: A FIC-AQP-NAM compressed image

1: **A FIC-AQP-NAM compressed file ($IMG_{C'}$) is taken. Set number of iteration (I_n) and scale factor (Sf) of image.**
2: **Image size and domain density are read as $size_x \times size_y$ and DD from the input file respectively.**
3: **The desired reconstructed image size $dsize_x \times dsize_y$ is calculated as-** $dsize_x \leftarrow Sf \times size_x$ **and** $dsize_y \leftarrow Sf \times size_x$
4: **Domain information is initialized that is similar with compression algorithm.**
5: **Take a black image (grey scale) of size $dsize_x \times dsize_y$ (say IMG_R).**
6: **Read all non-linear affine maps of IMG_C.**
7: **While ($I_n > 0$), do**
   7.1: Apply all non-linear affine maps of IMG_C over the image IMG_R that produces a completely new image.
   7.2: Now, this produced image acts as IMG_R.
   7.3: For each pixel value z_i of the image IMG_R, if $z_i < -128$, set $z_i \leftarrow$ -128 and if $z_i > 127$, set $z_i \leftarrow 127$.
   7.4: Set $I_n \leftarrow I_n - 1$.
   End While;
8: **Smooth the transition between adjacent ranges of IMG_R by pixel averaging.**
9: **Output $dsize_x \times dsize_y$ IMG_R as decompressed image.**
10: **Stop.**

6 The proposed variants

The compressed image file IMG_R of the proposed FIC-AQP-NAM technique keeps different parameters of a number of non-linear affine transforms. All these transforms have a number of parameter like scaling factor c_1, c_2, brightness offset o, position of domain (g_1, g_2) and rotation & flip operation types as shown in (3). Therefore, these parameters of the fractal compressed image can further be encoded to represent these in more compact forms in the compressed file by any loss-less technique without degradation of image quality. Hence, the size of the file may reduce and compression ratio may improve. Two variants of FIC-AQP-NAM have been suggested in this paper that use this concept. These variants apply optimality of LZW(OLZW) [27] and its modified version Modified OLZW (MOLZW) [28] that are effective variants of the dictionary based loss-less data compression technique LZW [46]. The proposed variants apply FIC-AQP-NAM with OLZW and MOLZW are termed as FIC-AQP-NAM-OLZW and FIC-AQP-NAM-MOLZW respectively. The encoding and decoding processes of FIC-AQP-NAM-OLZW are illustrated in Fig. 3a and b respectively. At the time of compression, it encodes the input image using proposed FIC-AQP-NAM compression process that produces a compressed file containing number of affine transforms. The parameters of affine transforms are further encoded by using OLZW technique to produce final compressed image. During decompression, the compressed image is reconstructed by FIC-AQP-NAM-OLZW decoder. First, the compressed file is decompressed by OLZW decoder to obtain parameters of collection of affine transforms. Then, the file containing affine parameters is decompressed by FIC-AQP-NAM decompression process to reconstruct the final decoded image.

The proposed variant FIC-AQP-NAM-MOLZW uses a different loss-less coding instead of OLZW that is known as MOLZW. The MOLZW is refined version of OLZW. The MOLZW encoding process is almost similar with OLZW except an additional checking to decide a dictionary is filled up or not before keeping a new phrase into the dictionary. If dictionary becomes filled up, least recently used phrase is eliminated to insert new phrase.

7 Result and analysis

All the experiments have been carried out on a machine with Intel i5 processor, DDR3 RAM (8 GB) and Ubuntu 14.04 OS. Five benchmark images (i.e. Lena, Pepper, Boats, Cameraman and Baboon) have been taken from an image database [45].

The results of the proposed FIC with AQP and non-linear affine maps (FIC-AQP-NAM) with its variants FIC-AQP-NAM-OLZW and FIC-AQP-NAM-MOLZW are presented against the existing FIC techniques that use Fisher's classification with QP (FICQP-CC) [9] and AQP (AFIC-CC) [29], FIC with AQP and archetype classification (AFIC-AC) [30], FIC with QP and DWT (FICQP-DWT) [43], adaptive edge based FIC using FFT (AE-FIC-FFT) and RDCT (AE-FIC-RDCT) [12], fast FIC method based on distance grouping (FFIC-DC)[22], FIC with AQP and hierarchical classification (AFIC-HC) [31] and it's modified scheme (AFIC-MHC) [38] and FIC with QP and non-linear affine maps (FICQP-NAM) [50]. The compression ratios (25) of all these methods for five images with average (say AVG) (27) and standard deviation (say SD) (28) are kept in Table 1. The compression ratio of Lena offered by the proposed FIC-AQP-NAM is better than all other examined techniques. For other four images, the rate of compression of the proposed FIC-AQP-NAM is higher than FICQP-CC, AE-FIC-FFT, AE-FIC-RDCT, FFIC-DC and FICQP-NAM techniques. As a result, the proposed FIC-AQP-NAM improves average

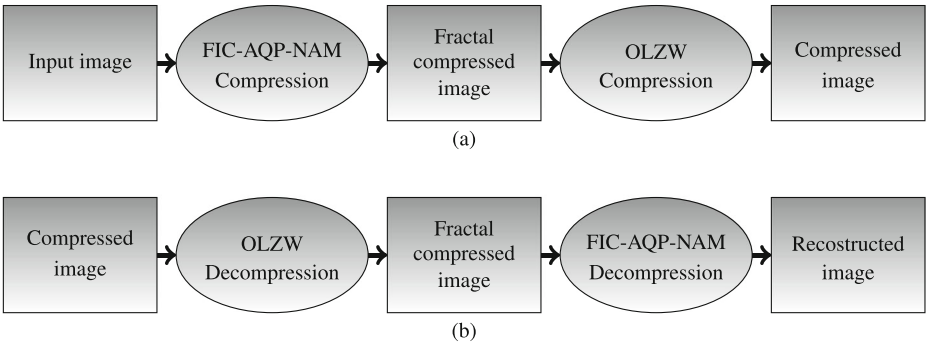

Fig. 3 The FIC-AQP-NAM-MOLZW technique **a** Compression process **b** Decompression process

compression ratio than FICQP-CC, AE-FIC-DWT, AE-FIC-FFT, AE-FIC-RDCT, FFIC-DC and FICQP-NAM techniques. The average space saving in % (26) of all the examined techniques are also shown in Fig. 4. For all five images, the the compression ratio of proposed variants FIC-AQP-NAM-OLZW and FIC-AQP-NAM-MOLZW are significantly better than other methods. The proposed variants FIC-AQP-NAM-OLZW and FIC-AQP-NAM-MOLZW yield 2.43% and 2.60 % increment of avg. compression ratio w.r.t. the base method FICQP-CC and 3.45% and 3.61% increment w.r.t. FICQP-NAM respectively. These improvements of compression ratio are achieved by encoding parameters of affine maps. As a result, the average space saving in % also increased of the variants.

$$Compression\ ratio = \frac{Size\ of\ Compressed\ file\ in\ bit}{Size\ of\ Original\ file\ in\ byte}\ bpp \tag{25}$$

$$Space\ saving = \left(1 - \frac{Size\ of\ Compressed\ file}{Size\ of\ Original\ file}\right) \times 100\% \tag{26}$$

Table 1 Compression ratio in bpp with average and standard deviation (DD=1, QLTY=1 and MIN=2 × 2

Compression technique	Lena	Pepper	Boats	Cameraman	Baboon	Avg	Std Dev
FICQP-CC [9]	1.360	1.026	1.119	1.278	1.383	1.233	0.155
AFIC-CC [29]	1.337	1.003	1.094	1.242	1.354	1.206	0.153
AFIC-AC [30]	1.338	1.003	1.098	1.241	1.363	1.207	0.153
FICQP-DWT[43]	1.342	1.005	1.122	1.254	1.366	1.218	0.153
AE-FIC-FFT [12]	1.375	1.031	1.124	1.279	1.398	1.241	0.160
AE-FIC-RDCT [12]	1.381	1.040	1.245	1.291	1.402	1.272	0.145
FFIC-DC [22]	1.348	1.030	1.105	1.242	1.356	1.216	0.145
AFIC-HC [31]	1.337	1.003	1.094	1.242	1.354	1.206	0.153
AFIC-MHC [38]	1.337	1.003	1.094	1.242	1.354	1.206	0.153
FICQP-NAM [50]	1.390	1.034	1.122	1.295	1.391	1.246	0.162
FIC-AQP-NAM	1.324	1.011	1.098	1.249	1.363	1.213	0.154
FIC-AQP-NAM-OLZW	1.330	1.002	1.093	1.240	1.352	1.203	0.152
FIC-AQP-NAM-MOLZW	1.324	1.002	1.091	1.238	1.350	1.201	0.150

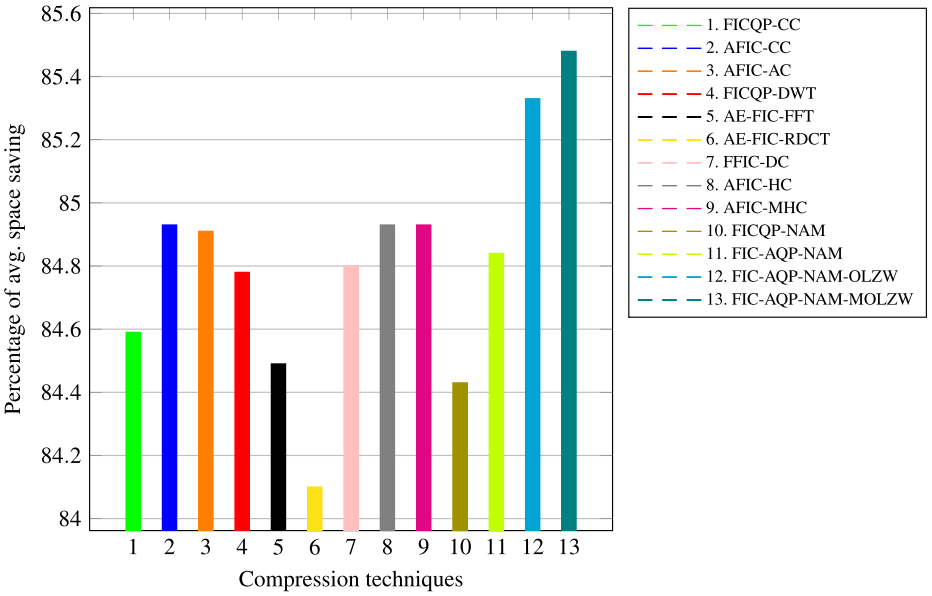

Fig. 4 The graphical representation of average percentage of space saving of all examined techniques

$$Average(\overline{x}) = \frac{1}{n}\sum_{i=0}^{n} x_i \tag{27}$$

$$Standard\ \ deviation = \sqrt{\frac{1}{n}\sum_{i=0}^{n} |(x_i - \overline{x})|} \tag{28}$$

During decompression process of the proposed technique, a gray scale black image is taken as the initial image where gray values of image are close to zero. The process converses within a finite number of iterations and reconstructs the image. The PSNRs [13, 40] of decoded images with AVG and SD of all experimented techniques are given in Table 2 where the PSNR is calculated by (29). Here, the RMS error between two images f and g as defined in (30) for image size $M \times N$.

$$PSNR = 20\log_{10}^{(\frac{255}{RMS})}\, dB \tag{29}$$

$$RMS = \sqrt{\frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N}(f_{i,j} - g_{i,j})^2} \tag{30}$$

The significant improvements of decoded image quality of all five images are noticed of the proposed methods. The proposed FIC-AQP-NAM and its variants improve 6.82 % avg. PSNR w.r.t. the base method FICQP-CC and 0.85% w.r.t. FICQP-NAM respectively. The PSNRs of proposed variants FIC-AQP-NAM-OLZW and FIC-AQP-NAM-MOLZW are same with FIC-AQP-NAM and there are no degradations of the PSNR for increasing compression ratios. The reconstructed Lena images after 2^{nd}, 4^{th}, 6^{th},and 8^{th} iterations of the proposed FIC-AQP-NAM are illustrated in Fig. 5a, b, c and d respectively where

Table 2 PSNR in dB with average and standard deviation (DD=1, QLTY=1 and MIN=2 × 2

Compression technique	Lena	Pepper	Boats	Cameraman	Baboon	Avg	Std Dev
FICQP-CC [9]	28.90	29.83	25.30	27.29	20.11	26.286	3.857
AFIC-CC [29]	29.31	30.06	26.42	27.83	21.58	27.040	3.358
AFIC-AC [30]	29.36	30.09	26.48	27.85	21.59	27.074	3.367
FICQP-DWT [43]	28.86	29.81	25.26	27.27	20.10	26.260	3.851
AE-FIC-FFT [12]	29.20	29.85	26.23	27.47	21.35	26.820	3.374
AE-FIC-RDCT [12]	29.23	30.04	26.20	27.55	21.42	26.888	3.400
FFIC-DC [22]	29.35	30.08	26.52	27.87	21.58	27.080	3.367
AFIC-HC [31]	29.31	30.06	26.42	27.83	21.58	27.040	3.358
AFIC-MHC [38]	29.31	30.06	26.42	27.83	21.58	27.040	3.358
FICQP-NAM [50]	31.70	31.82	26.45	27.86	21.39	27.844	4.309
FIC-AQP-NAM	31.78	32.11	26.68	27.97	21.86	28.080	4.201
FIC-AQP-NAM-OLZW	31.78	32.11	26.68	27.97	21.86	28.080	4.201
FIC-AQP-NAM-MOLZW	31.78	32.11	26.68	27.97	21.86	28.080	4.201

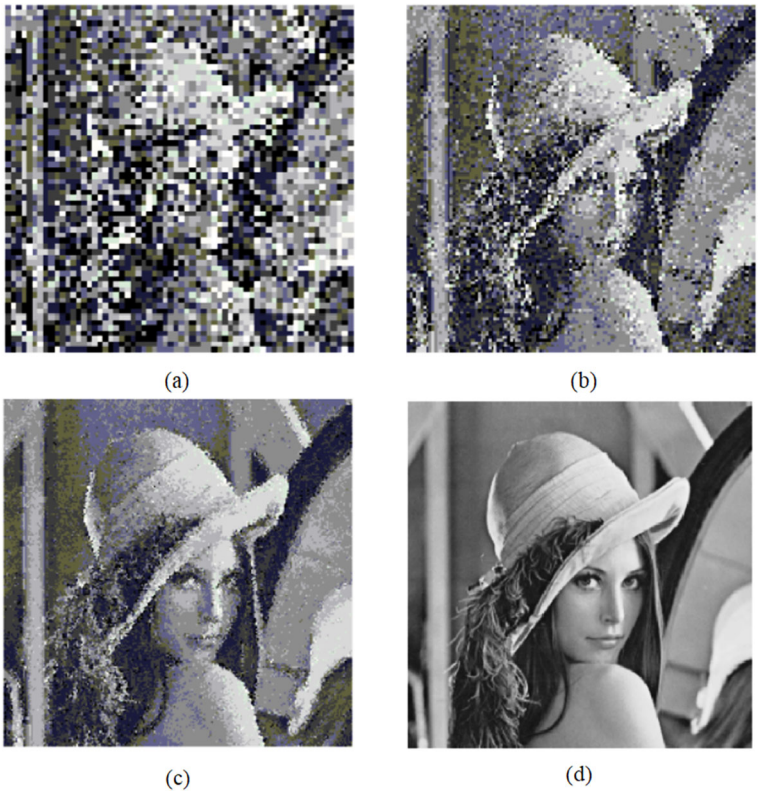

Fig. 5 Reconstructed images after (a) 2^{nd}, (b) 4^{th}, (c) 6^{th}, (d) 8^{th} iterations using proposed FIC-AQP-NAM

image becomes more close with original image after each iteration. The original benchmark images, reconstructed images using FICQP-CC, FICQP-NAM and proposed FIC-AQP-NAM/ FIC-AQP-NAM-OLZW/ FIC-AQP-NAM-MOLZW are illustrated in Fig. 6a, b, c and d respectively. Another image quality measurement metric i.e. Structural Similarity Index Measure (SSIM) [13, 41] of these reconstructed images are also measured to compare image quality of proposed methods with FICQP-CC and FICQP-NAM where it lies between 0 and 1 and higher value means better image quality. The SSIM values of decoded five images for FICQP-CC, FICQP-NAM and proposed FIC-AQP-NAM techniques are mentioned in the Fig. 6. The SSIM values of decoded Lena image for FICQP-CC, FICQP-NAM and proposed techniques are 0.7725, 0.8520 and 0.8645 respectively. The same for Pepper image are 0.7958, 0.8555 and 0.8627 respectively. The decoded Boats image yields 0.6825, 0.7212 and 0.7270 and Cameraman image yields 0.7322, 0.7565 and 0.7592 for FICQP-CC, FICQP-NAM and proposed methods. The SSIM values of decoded Baboon image for FICQP-CC, FICQP-NAM and proposed methods are 0.5525, 0.5847 and 0.5965 respectively. Hence, the image qualities of the proposed techniques in terms of SSIM are significantly better than its counterparts. The distance between histogram of original and reconstructed image using intersection [25], correlation [37], χ^2-test [48], Kullback-Leibler distance [17] and Bhattacharyya distance[5] metrics are given in Table 3. For higher values of intersection and correlation, better matching of original and reconstructed images are found. However, the lower values of other metrics indicate better matching of these images. If the contractive conditions of affine maps are not restricted, the decoding process does not converge. Fig. 7a is the decoded Lena image after 8^{th} iterations without restriction of contractive conditions and it is totally not recognizable. This result shows the importance of contractive condition. The decoded Lena images without adjusting the gray values of image are shown in Fig. 7b after 8^{th} iteration where decoding starts with a 256×256 black gray scale image. Here also, the decoding process does not converse within a finite number of iterations. The reconstructed images have a large numbers of blocky portions. Therefore, the gray value adjustment is another important requirement.

Table 4 contains the time of encoding in second with AVG and SD of all these methods. It is observed that the avg. encoding time of the proposed FIC-AQP-NAM is lower than AFIC-CC and AFIC-AC but other examined techniques. Because,the compression process takes more time to find optimum parameters of non-linear affine maps. The adaptive quadtree partitioning also makes the compression slow where partitioning points are determined based on image context.The two variants become slightly slow than FIC-AQP-NAM also since these two applied OLZW or MOLZW after FIC. The decompression times of all the methods examined with AVG and SD are given in Table 5. It is noticed that the decompression times of these methods are low than corresponding compression times since the decompression process iterates only affine transforms on an image to reconstruct the decoded image. However, the decompression times of proposed methods are greater than other examined methods since it uses non-linear affine map. Both the proposed variants have more decoding time for additional OLZW/MOLZW decoding step.

Figure 8 show how the avg. quality (i.e. avg. PSNRs) of five images vary with avg. compression ratio for FICQP-CC, FICQP-NAM, proposed FIC-AQP-NAM and its two variants. The avg. PSNR values are increased with the increasing avg. compression ratio of all five techniques since better PSNR can be obtained if the compressed image keeps more affine transforms. As a result, the size of the compressed image becomes larger offering high compression ratio. It is also noticed that all three proposed techniques offer higher avg. PSNRs with respected to avg. compression ratio than its counter parts and FIC-AQP-NAM is better among those. Figure 9a and b depict how the qualities (i.e. PSNRs) of Lena image vary with

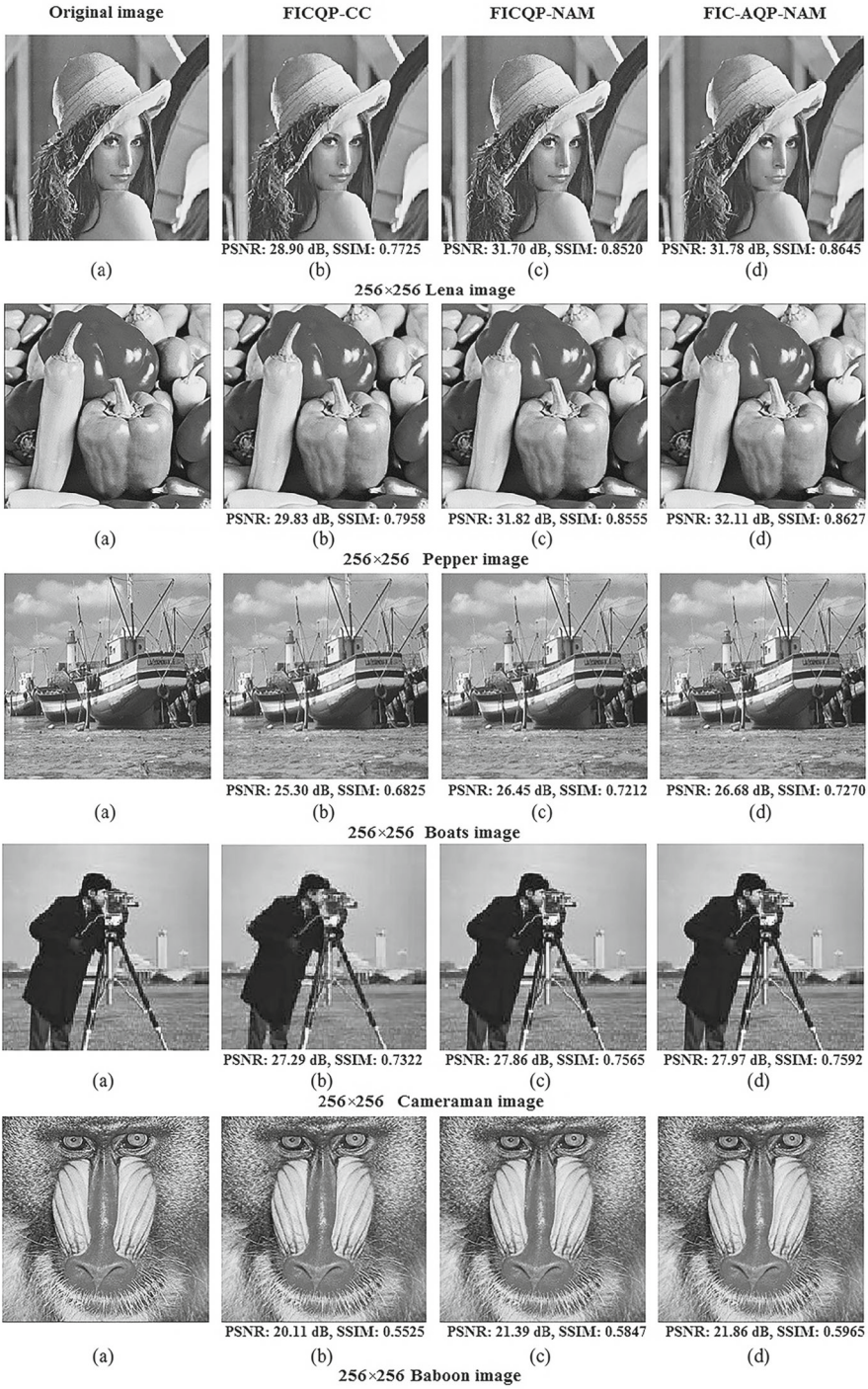

Fig. 6 Images **a** Original and reconstructed using techniques **b** FICQP-CC **c** FICQP-NAM **d** proposed FIC-AQP-NAM and its variants

Table 3 The comparison of histogram between original and recontrscted image of the proposed FIC-AQP-NAM technique

Metric	Lena	Pepper	Boats	Cameraman	Baboon
Intersection [25]	102.26	98.53	95.30	33.94	86.42
Correlation [37]	0.992	0.974	0.942	0.967	0.873
χ^2-test [48]	0.382	0.610	2.482	4.952	2.540
Kullback-Leibler distance [17]	3.066	6.010	5.981	9.587	12.127
Bhattacharyya distance [5]	0.022	0.020	0.263	0.049	0.240

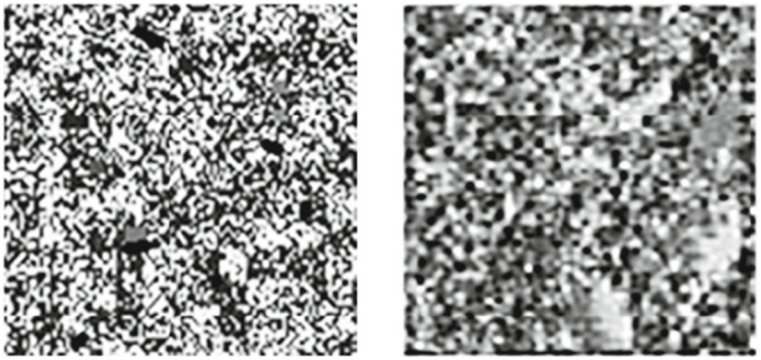

Fig. 7 The decoded Lena image after 8^{th} iteration **a** without restriction of contractive condition **b** without gray value adjustments

Table 4 Compression time in second with average and standard deviation (DD=1, QLTY=1 and MIN=2 × 2)

Compression technique	Lena	Pepper	Boats	Cameraman	Baboon	Avg	Std Dev
FICQP-CC [9]	0.22	0.23	0.20	0.28	0.23	0.232	0.029
AFIC-CC [29]	0.36	0.34	0.38	0.46	0.33	0.374	0.052
AFIC-AC [30]	0.34	0.33	0.35	0.44	0.32	0.356	0.048
FICQP-DWT [43]	0.17	0.15	0.16	0.19	0.20	0.174	0.021
AE-FIC-FFT [12]	0.21	0.21	0.18	0.25	0.20	0.210	0.025
AE-FIC-RDCT [12]	0.20	0.21	0.18	0.24	0.20	0.206	0.022
FFIC-DC [22]	0.19	0.21	0.18	0.24	0.20	0.204	0.023
AFIC-HC [31]	0.16	0.14	0.15	0.17	0.16	0.156	0.011
AFIC-MHC [38]	0.16	0.14	0.15	0.16	0.15	0.152	0.008
FICQP-NAM [50]	0.31	0.32	0.30	0.39	0.31	0.326	0.036
FIC-AQP-NAM	0.34	0.31	0.35	0.41	0.32	0.346	0.039
FIC-AQP-NAM-OLZW	0.36	0.34	0.38	0.47	0.34	0.378	0.054
FIC-AQP-NAM-MOLZW	0.37	0.34	0.39	0.47	0.35	0.384	0.052

Table 5 Decompression time in second with average and standard deviation (DD=1, QLTY=1 and MIN=2 × 2)

Compression technique	Lena	Pepper	Boats	Cameraman	Baboon	Avg	Std Dev
FICQP-CC [9]	0.024	0.023	0.024	0.027	0.023	0.024	0.002
AFIC-CC [29]	0.032	0.031	0.040	0.042	0.031	0.035	0.005
AFIC-AC [30]	0.037	0.032	0.044	0.041	0.032	0.037	0.005
FICQP-DWT [43]	0.034	0.033	0.041	0.042	0.032	0.036	0.005
AE-FIC-FFT [12]	0.025	0.023	0.025	0.028	0.024	0.025	0.002
AE-FIC-RDCT [12]	0.025	0.018	0.019	0.021	0.022	0.021	0.003
FFIC-DC [22]	0.02	0.015	0.018	0.017	0.016	0.017	0.002
AFIC-HC [31]	0.023	0.019	0.018	0.023	0.026	0.022	0.003
AFIC-MHC [38]	0.025	0.021	0.024	0.022	0.024	0.023	0.002
FICQP-NAM [50]	0.025	0.025	0.024	0.028	0.023	0.025	0.002
FIC-AQP-NAM	0.034	0.032	0.041	0.042	0.032	0.036	0.005
FIC-AQP-NAM-OLZW	0.036	0.037	0.043	0.043	0.034	0.039	0.004
FIC-AQP-NAM-MOLZW	0.037	0.037	0.044	0.043	0.035	0.039	0.004

time of compression and decompression respectively. The improvements of PSNR values are noticed with the increasing of both compression and decompression time of all five techniques since more time is required to compress and decompress images with large number of affine transforms for better PSNRs. The variations of compression times and compression ratios for Lena image with increasing mean square error (MSE) are illustrated in Fig. 10a

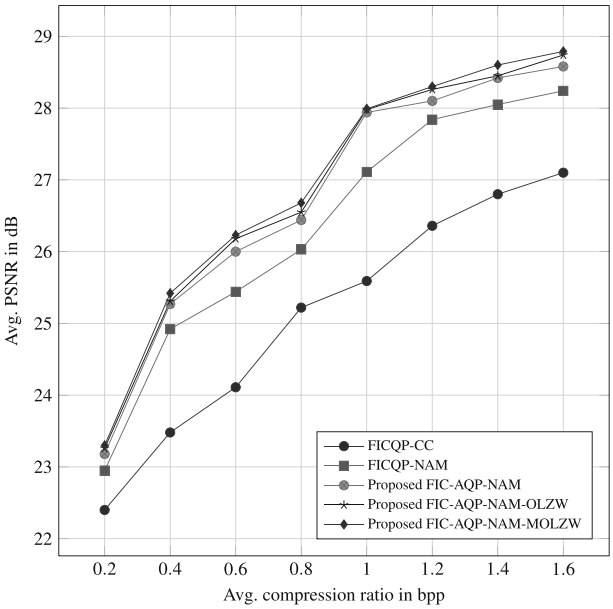

Fig. 8 The variation of avg. PSNR (in dB) with increasing avg. compression ratio (in bpp)

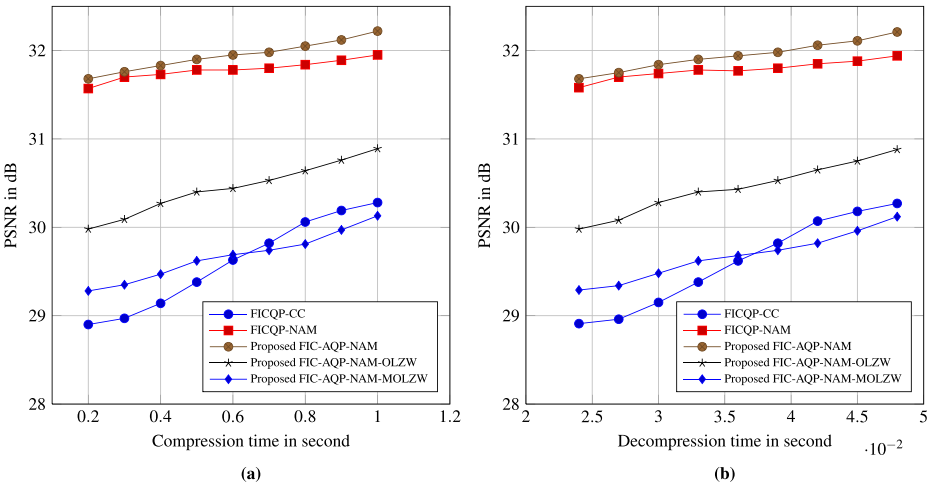

Fig. 9 **a** The variation of PSNR (in dB) with increasing compression time (in second) for Lena image **b** The variation of PSNR (in dB) with increasing decompression time (in second) for Lena image

and b respectively of the same five techniques. It is observed that the compression times of these techniques decrease slowly and compression ratio increases significantly with the increasing MSE. Low variation of compression time indicating these techniques execute properly for any image quality and higher MSE yields poor quality reconstructed image to lead high ratio of compression. Figure 11a shows the plots of compression time of Lena image with DD of FICQP-CC, FICQP-NAM, proposed FIC-AQP-NAM and its two variants. It is observed that the compression times of these techniques drop with higher domain density since number of domain searching reduces for higher domain density. The variation

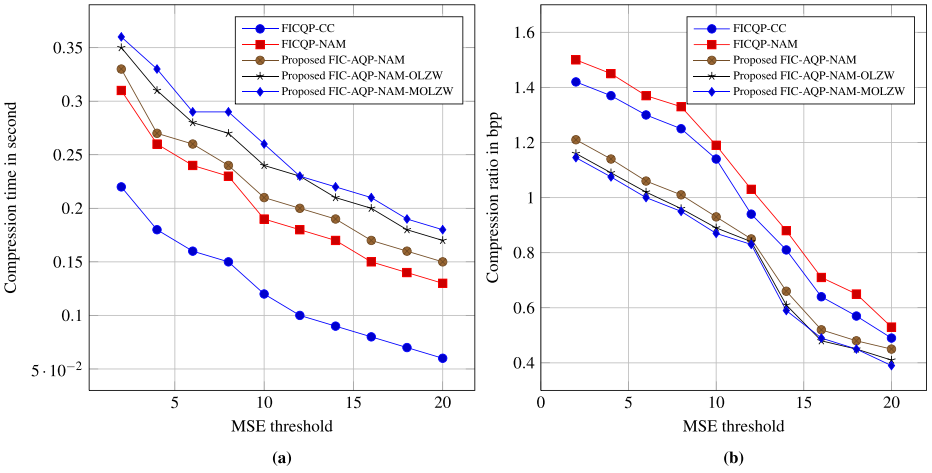

Fig. 10 **a** The variation of compression time (in second) with increasing MSE threshold for Lena image **b** The variation of compression ratio (in bpp) with increasing MSE threshold for Lena image

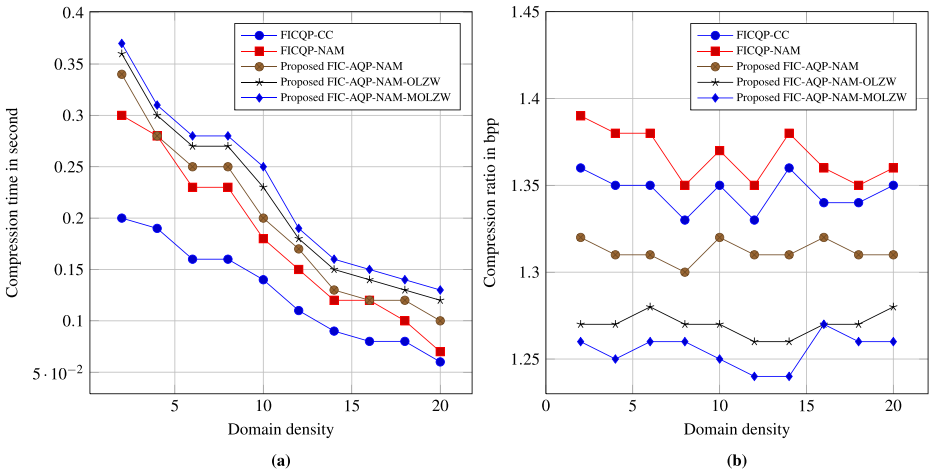

Fig. 11 **a** The variation of compression time (in second) with increasing domain density for Lena image **b** The variation of compression ratio (in bpp) with increasing domain density for Lena image

of compression ratio of Lena image with domain density is shown in Fig. 11b where a very low variation in compression ratios of these techniques are noticed for increasing domain density. It is observed from Fig. 12a and b that the decompression times of these techniques decrease with the increasing MSE but no significant variation for DD. Figure 13a and b represent the plots for variation of PSNR and SSIM with MSE respectively where both PSNR and SSIM decrease with increasing MSE of the same five techniques. Because, increasing MSE decreases image quality.

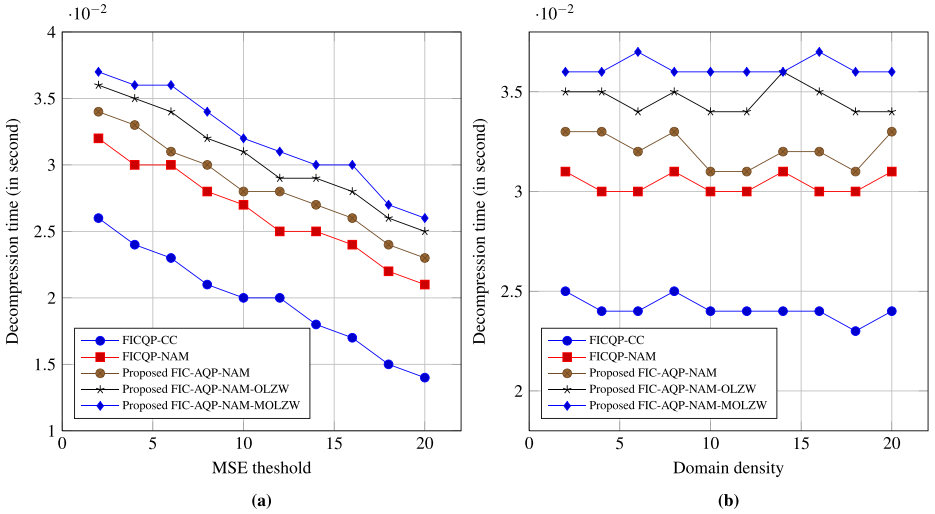

Fig. 12 **a** The variation of decompression time (in second) with increasing MSE theshold **b** The variation of decompression time (in second) with increasing domain density for Lena image

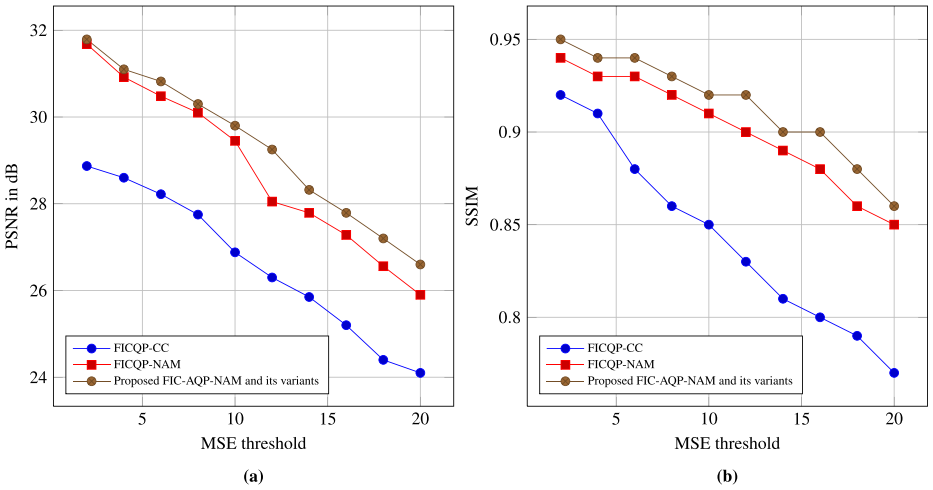

Fig. 13 **a** The variation of PSNR with increasing MSE threshold for Lena image **b** The variation of SSIM with increasing MSE threshold for Lena image

8 Conclusion

The proposed technique greatly improves the decoded image quality than its counterparts by using non-linear affine maps to produce better pixel intensity approximations and adaptive quadtree partitioning scheme to split range block in such a way that it enhance the possibility of finding self-matching portions. The technique also offers better or at least almost close compression rate compare with existing techniques. Two variants further improve compression ratio significantly using loss-less coding keeping image quality unchanged. However, the requirement of gray values adjustment after end of each an iteration to converse decompression process is extra overhead of the proposed technique. Further investigation should be continued to overcome this limitation.

Acknowledgements We'd like to thank to the Dept. of Computer Science, Vidyasagar University, Paschim Medinipur for providing infrastructure.

References

1. Al-Jawfi R, Al-Helali B, Ahmed A (2014) Fractal image compression using self-organizing mapping. Appl Math 05:1810–1819
2. Antonini M, Barlaud M, Mathieu P, Daubechies I (1992) Image coding using wavelet transform. IEEE Trans Image Process 1(2):205–220
3. Barnsley MF (1993) Fractals Everywhere, 2nd edn, Academic Press, New York
4. Bhattacharya N, Roy S, Nandi U, Banerjee S (2015) Fractal image compression using hierarchical classification of sub-images. In: 10th International conference on computer vision theory and applications (VISAPP 2015), SCITEPRESS, pp 46–53
5. Bhattachayya A (1943) On a measure of divergence between two statistical populations defined by their probability distributions. In: Bulletin of the calcutta mathematical society, pp 99–109
6. Bobde SS, Kulkarni MV, Kulkarni PV (2010) Fractal image compression using genetic algorithm. In: 2010 International conference on advances in computer engineering. IEEE, pp 241–243

7. Bouchemel A, Abed D, Moussaoui A (2018) Enhancement of compressed image transmission in WMSNs using modified μ -nonlinear transformation. IEEE Commun Lett 22(5):934–937
8. DeVore RA, Jawerth B, Lucier BJ (1992) Image compression through wavelet transform coding. IEEE Trans Inf Theory 38(2):719–746
9. Fisher Y (1995) Fractal image compression: theory and application. Springer Verlag, New York
10. Fu C, Zhu Z (2009) A DCT-based fractal image compression method. In: International workshop on chaos-fractals theories and applications. IEEE, pp 439–443
11. Gupta R, Mehrotra D, Tyagi RK (2016) Adaptive searchless fractal image compression in DCT domain. Imaging Sci J 64(7):374–380
12. Gupta R, Mehrotra D, Tyagi RK (2018) Comparative analysis of edge-based fractal image compression using nearest neighbor technique in various frequency domains. Alex Eng J 57(3):1525–1533
13. Hore A, Ziou D (2010) Image quality metrics: PSNR vs. SSIM. In: International conference on pattern recognition proceedings. IEEE, pp 2366–2369
14. Jacquin AE (1993) Fractal image coding: a review. Proc IEEE 81(10):1451–1465
15. Jeng J, Tseng C, Hsieh J (2009) Study on huber fractal image compression. IEEE Trans Image Process 18(5):995–1003
16. Kocic LM, Matejic MM (2006) Contractive affine transformations of complex plane and its applications. Ser Math Inform 21:65–75
17. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Statist 22(1):79–86
18. Lakshmi GVM (2016) Implementation of image compression using fractal image compression and neural networks for MRI images. In: 2016 International conference on information science (ICIS). IEEE, pp 60-64
19. Li Y (2016) Collision analysis and improvement of a hash function based on chaotic tent map. Optik (Stuttg.) 127(10):4484–4489
20. Li Y, Ge G (2019) Cryptographic and parallel hash function based on cross coupled map lattices suitable for multimedia communication security. Multimed Tools Appl 78(13):17,973–17,994
21. Liu S, Fu W, Liqiang H, et al. (2017a) Distribution of primary additional errors in fractal encoding method. Multimed Tools Appl 76:5787–5802
22. Liu S, Pan Z, Cheng X (2017b) A novel fast fractal image compression method based on distance clustering in high dimensional sphere surface. Fractals 25(4):1740,004–1–1740,004–11
23. Liu S, Zhang Z, Qi L, et al. (2016) A fractal image encoding method based on statistical loss used in agricultural image compression. Multimed Tools Appl 75:15,525–15,536
24. Lu J, Ye Z, Zou Y (2013) Huber fractal image coding based on a fitting plane. IEEE Trans Image Process 22(1):134–145
25. Name MH, Lima J, Boff F, Filho D, Falate R (2012) Histogram comparison using intersection metric applied to digital images analysis. Iberoamerican J Appl Comput 2:11–18
26. Nandi U, Mandal JK (2012) Fractal image compression using fast context independent HV partitioning scheme. In: 2012 International symposium on electronic system design (ISED), IEEE, pp 306-308
27. Nandi U, Mandal JK (2012) A compression technique based on optimality of LZW code (OLZW). In: 2012 Third international conference on computer and communication technology. IEEE, pp 166–170
28. Nandi U, Mandal JK (2013) Modified compression techniques based on optimality of LZW code (MOLZW). Procedia Technology 10:949–956
29. Nandi U, Mandal JK (2015) Fractal image compression with adaptive quadtree partitioning and lossless encoding on the parameters of affine transformations. In: Mandal JK, et al. (eds) Information systems design and intelligent applications, advances in intelligent systems and computing. Springer, New Delhi, pp 73–83
30. Nandi U, Mandal JK (2016) Efficiency of adaptive fractal image compression with archetype classification and its modifications. Int J of Comput and Appl 38(2-3):156–163
31. Nandi U, Mandal JK (2018) A novel hierarchical classification scheme for adaptive quadtree partitioning based fractal image coding. In: Mandal JK, Sinha D (eds) Social transformation – digital way. Springer, Singapore, pp 603-615
32. Nelson M (2008) The data compression book, 2nd edn. BPB Publications, India
33. Nodehi A, Sulong G, Al-Rodhaan M, et al. (2014) Intelligent fuzzy approach for fast fractal image compression. Eurasip J Adv Signal Process, 2014:112
34. Roy S, Kumar S, Chanda B, et al. (2017) Fractal image compression using upper bound on scaling parameter. Chaos Solitons Fract 106:16–22
35. Said A, Pearlman WA (1996) A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans Circuits Syst Video Technol 6(3):243–250
36. Shapiro JM (1993) Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans on Signal Process 41(12):3445–3462

37. Suneja A, Walia E (2010) A conceptual study on image matching techniques. Global J Comput Sci Tech 10(12):83–88
38. Utpal N (2019) An adaptive fractal-based image coding with hierarchical classification strategy and its modifications. Innov Syst Softw Eng 15:35–42
39. Wallace GregoryK (1999) The JPEG still picture compression standard. Commun ACM 34:31–44
40. Wang Q, Bi S (2016) Prediction of the PSNR quality of decoded images in fractal image coding. Math Probl Eng 2016:1–13
41. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
42. Wang XY, Wang YX, Yun JJ (2010) An improved no-search fractal image coding method based on a fitting plane. Image Vis Comput 28(8):1303–1308
43. Wang X, Zhang D (2014) Discrete wavelet transform-based simple range classification strategies for fractal image coding. Nonlinear Dyn 75:439–448
44. Wang J, Zheng N (2013) A novel fractal image compression scheme with block classification and sorting based on pearson's correlation coefficient. IEEE Trans Image Process 22:3690–3702
45. Weber G (1993) Usc-sipi image database: version 4. Dept Elect Eng-Syst. Tech Rep. Univ Southern California, Los Angeles, CA, USA
46. Welch (1984) A technique for high-performance data compression. Computer 17(6):8–19
47. Xing C, Ren Y, Li X (2008) A hierarchical classification matching scheme for fractal image compression. In: 2008 Congress on image and signal processing. IEEE, pp 283–286
48. Yates F (1934) Contingency tables involving small numbers and the χ^2 test. J R Stat Soc (Supplement) 1(2):217–235
49. Zhang Y, Li Y, Wen W, Wu Y, Chen JX (2015) Deciphering an image cipher based on 3-cell chaotic map and biological operations. Nonlinear Dyn 82(4):1831–1837
50. Zhao Y, Yuan B (1998) A new affine transformation: its theory and application to image coding. IEEE Trans Circuits Syst Video Technol 8(3):269–274

Utpal Nandi received his M.Sc. in Computer Science from Vidyasagar University, Midnapore, Paschim Medinipur, West Bengal, India in 2006. He did his M.Tech in Computer Science & Engineering in 2009 and Ph.D. in 2018 from University of Kalyani, Nadia, West Bengal, India. He is currently working as an Assistant Professor in the Dept. of Computer Science, Vidyasagar University, Midnapore, Paschim Medinipur, West Bengal, India. He has 9 years of teaching and research experiences. His research interest includes data and image compression techniques, image processing and multimedia technology. He has published more than 23 papers in international journals, book chapters and conferences.